DETAILED ACTION

Continued Examination Under 37 CFR 1.114

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 26 January 2010 has been entered.

Claim Rejections - 35 USC § 102

1. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

2. Claims 1-3, 7, 11-13, 17, 21-23, 27, 31-33, 37, 41-43, 47 are rejected under 35 U.S.C. 102(e) as being anticipated by Robinett et al. (PG Pub US 2002/0126711 A1).

Regarding claims **1 and 11**, Robinett et al. discloses a multiplexing apparatus which multiplexes a plurality of audio elementary data streams and a plurality of video elementary data streams to generate one multiplexed stream (fig. 1), the multiplexing apparatus comprising:

encoders receiving video data from a video source, receiving audio data from an audio source, encoding the video data into the video elementary data streams, encoding the audio data into the audio elementary data streams, dividing the video elementary data streams into a plurality of video data units and dividing the audio elementary streams into a plurality of audio data units ("TSs are bit streams that contain the data of one or more compressed/encoded audiovideo programs. Each TS is formed as a sequence of fixed length transport packets. Each compressed program includes data for one or more compressed elementary streams (ESs), such as a digital video signal and/or a digital audio signal. The transport packets also carry program clock references (PCRs) for each program, which are time stamps of an encoder system time clock to which the decoding and presentation of the respective program is synchronized" [0033], "one or more data injection sources 50 and one or more data extraction destinations 60. These sources 50 and destinations 60 may themselves be implemented as PC compatible computers. However, the sources 50 may also be devices such as cameras, video tape players, communication demodulators/receivers and the destinations may be display monitors, video tape recorders, communications modulators/transmitters, etc. The data injection sources 50 supply TS, ES or other data to the remultiplexer 30, e.g., for remultiplexing into the outputted TSs TS4 and/or TSS" [0066]);

a memory which stores the plurality of video and audio data units that are composed of an arbitrary amount of said video and audio elementary data streams ("The use of the cache 114 enables transport packets to be received and stored or to be

retrieved and outputted" [0074] and further "TSs are bit streams that contain the data of one or more compressed/encoded audio-video programs. Each TS is formed as a sequence of fixed length transport packets" [0033]);

an instruction generating means for calculating an order of multiplexing the video data units and the audio data units based on storage locations supplied by the encoders ("The descriptors are maintained in order of receipt in the receipt queue. In the case of outputting transport packets from an output port, the data link control circuit sequentially retrieves each descriptor from the transmit queue, and the transport packet to which each retrieved descriptor points" [0037], "When an adaptor is used to input transport streams, the data link control circuit allocates to each received transport packet to be retained, an unused descriptor in one of a sequence of descriptor storage locations, of a queue allocated to the input port" [0035]), for generating an instruction set of a plurality of multiplexing instruction data which describe the storage location and order of multiplexing of each data unit ("The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the

sequence in order" [0036]) and for storing the instruction set into the memory ("The cache 114 also stores descriptor data for each transport packet" [0074], "The memory can store one or more queues of descriptor storage locations, such as a queue assigned to an input port and a queue assigned to an output port" [0034]); and

a multiplexed stream generating means for reading the instruction set from the memory ("When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the sequence in order" [0036]), for generating one multiplexed stream by reading the data units from the memory in a predetermined order based upon the instruction set ("The data link control circuit also retrieves from the cache the transport packers stored in transport packet storage locations to which the retrieved descriptors point. The data link control circuit outputs each retrieved transport packet in a unique time slot (i.e., one transport packet per time slot) of a transport stream outputted from the output port. The direct memory access circuit obtains from the memory for storage in the cache, descriptors of the queue assigned to the output port in storage locations following the descriptor storage locations in which a last cached descriptor of the sequence is stored. The direct memory access circuit also obtains each transport packet stored in a transport packet location to which the obtained descriptors point" [0036]) and outputting the data units corresponding to the instruction set ("combine program information of two TSs, namely, TS1 and TS2, into a third TS, namely, TS3" [0093]; "TS1 illustratively is received at a first adaptor 110, TS2 illustratively is received

at a second adaptor 110 and TS3 illustratively is transmitted from a third adaptor 110 of the same remultiplexer node 100" [0093]; "the processor 160 examines descriptors from the receipt queues (and/or possibly other queues containing descriptors of transport packets not yet scheduled for output) and identifies up to j≥1 descriptors pointing to transport packets to be outputted from the interrupting adaptor 110"; and further "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the transport packet in the transport packet storage location to which the descriptor points" [0144]).

Regarding claims **2 and 12**, Robinett et al. discloses a multiplexing apparatus which multiplexes a plurality of video elementary data streams and a plurality of audio elementary data streams to generate one multiplexed stream (fig. 1), the multiplexing apparatus comprising:

encoders receiving video data from a video source, receiving audio data from an audio source, encoding the video data into the video elementary data streams, encoding the audio data into the audio elementary data streams, dividing the video elementary data streams into a plurality of video data units and dividing the audio elementary streams into a plurality of audio data units ("TSs are bit streams that contain the data of one or more compressed/encoded audiovideo programs. Each TS is formed as a sequence of fixed length transport packets. Each compressed program includes data for one or more compressed elementary streams (ESs), such as a digital video signal and/or a digital audio signal. The transport packets also carry program clock references (PCRs) for each program, which are time stamps of an encoder system time

clock to which the decoding and presentation of the respective program is synchronized" [0033], "one or more data injection sources 50 and one or more data extraction destinations 60. These sources 50 and destinations 60 may themselves be implemented as PC compatible computers. However, the sources 50 may also be devices such as cameras, video tape players, communication demodulators/receivers and the destinations may be display monitors, video tape recorders, communications modulators/transmitters, etc. The data injection sources 50 supply TS, ES or other data to the remultiplexer 30, e.g., for remultiplexing into the outputted TSs TS4 and/or TSS" [0066]);

a memory which stores plurality of video and audio data units that are composed of an arbitrary amount of said video and audio elementary data streams ("The use of the cache 114 enables transport packets to be received and stored or to be retrieved and outputted" [0074] and further "TSs are bit streams that contain the data of one or more compressed/encoded audio-video programs. Each TS is formed as a sequence of fixed length transport packets" [0033]);

an instruction generating means for calculating an order of multiplexing the video data units and audio data units based on storage locations supplied by the encoders ("The descriptors are maintained in order of receipt in the receipt queue. In the case of outputting transport packets from an output port, the data link control circuit sequentially retrieves each descriptor from the transmit queue, and the transport packet to which each retrieved descriptor points" [0037], "When an adaptor is used to input transport streams, the data link control circuit allocates to each received transport packet to be

retained, an unused descriptor in one of a sequence of descriptor storage locations, of a queue allocated to the input port" [0035]), generating an instruction set of a plurality of multiplexing instruction data which describe the storage location and order of multiplexing of each data unit while generating command instruction data having stated therein an instruction for execution of a data processing to be executed in an arbitrary position in the multiplexing instruction data ("The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076]; and "each receipt descriptor has a field 129-9 in which information pertinent to scrambling or descrambling can be stored, such as the control word to be used in scrambling the transport packet or a pointer to the appropriate control word table containing control words for use in scrambling or descrambling the transport packet" [0165], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the sequence in order" [0036]), and storing the instruction set and command instruction data into the memory ("The cache 114 also stores descriptor data for each transport packet" [0074], "The memory can store one or more queues of

descriptor storage locations, such as a queue assigned to an input port and a queue assigned to an output port" [0034]);

a multiplexed stream generating means for reading the instruction set from the memory, for generating one multiplexed stream including the video and audio elementary data streams and command data by reading the data units and command instruction data from the memory in a predetermined order based upon the instruction set and for outputting the data units corresponding to the instruction set, after reading the instruction set ("combine program information of two TSs, namely, TS1 and TS2, into a third TS, namely, TS3" [0093]; "TS1 illustratively is received at a first adaptor 110, TS2 illustratively is received at a second adaptor 110 and TS3 illustratively is transmitted from a third adaptor 110 of the same remultiplexer node 100" [0093]; "the processor 160 examines descriptors from the receipt queues (and/or possibly other queues containing descriptors of transport packets not yet scheduled for output) and identifies up to j≥1 descriptors pointing to transport packets to be outputted from the interrupting adaptor 110"; and further "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the transport packet in the transport packet storage location to which the descriptor points" [0144]), or by outputting command data having stated therein the execution instruction stated in the command instruction data, after reading the command instruction data ("In processing descriptors and transport packets, the descrambler 115 uses the PID of the transport packet, to which the currently examined descriptor points, to index a descrambling map located in the cache 114" [0172] and

"The indexed entry of the descrambling map indicates whether or not the transport packet is scrambled and, if scrambled, one or more control words that can be used to descramble the transport packet. The indexed entry of the descrambling map can contain the control words corresponding to the PID of the transport packet or a pointer to a memory location in which the respective control word is stored" [0172]); and

a command executing means which is supplied with a multiplexed stream output from the multiplexed stream generating means and makes a processing corresponding to an instruction content stated in the command data when a data row in the multiplexed stream is command data ("If the indexed entry of the descrambling map indicates that the transport packet is to be descrambled, the descrambler 115 obtains the control word corresponding to the PID of the transport packet and descrambles the transport packet data using the control word" [0172]), or outputs the multiplexed stream as it is when the data row in the input multiplexed stream is video and audio elementary data stream ("If the indexed entry of the descrambling map indicates that the transport packet to which the accessed descriptor points is not to be descrambled, the descrambler 115 simply sets the status bit(s) 129-7 of the descriptor to indicate that the next processing step, according to the order of the defined sequence of processing steps, may be performed on the descriptor and transport packet to which it points" [0172]).

Regarding claims **21, 31, 41**, Robinett discloses an apparatus for multiplexing a plurality of video elementary data streams and a plurality of audio elementary data streams to generate a multiplexed stream (figs. 1-2):

a bus (bus 24, fig. 1 or bus 130, fig. 2);

encoders receiving video data from a video source, receiving audio data from an audio source, encoding the video data into the video elementary data streams, encoding the audio data into the audio elementary data streams, dividing the video elementary data streams into a plurality of video data units and dividing the audio elementary streams into a plurality of audio data units ("TSs are bit streams that contain the data of one or more compressed/encoded audiovideo programs. Each TS is formed as a sequence of fixed length transport packets. Each compressed program includes data for one or more compressed elementary streams (ESs), such as a digital video signal and/or a digital audio signal. The transport packets also carry program clock references (PCRs) for each program, which are time stamps of an encoder system time clock to which the decoding and presentation of the respective program is synchronized" [0033], "one or more data injection sources 50 and one or more data extraction destinations 60. These sources 50 and destinations 60 may themselves be implemented as PC compatible computers. However, the sources 50 may also be devices such as cameras, video tape players, communication demodulators/receivers and the destinations may be display monitors, video tape recorders, communications modulators/transmitters, etc. The data injection sources 50 supply TS, ES or other data to the remultiplexer 30, e.g., for remultiplexing into the outputted TSs TS4 and/or TSS" [0066]);

a data memory linked to the bus storing the plurality of video and audio elementary data streams (the following elements either alone or in combination of

processor 21, memory 23, disk memory 25, storage 40, fig. 1 or cache 114, host memory 120, fig. 2);

an instruction memory linked to the bus storing an instruction set (the following elements either alone or in combination of processor 21, memory 23, disk memory 25, storage 40, fig. 1 or cache 114, host memory 120, fig. 2);

a CPU linked to the bus for generating the instruction set having stated therein a storage location in the data memory of a data unit formed from an elementary stream and an order of multiplexing the data units and for storing the generated instruction set into the instruction memory ("The purpose of the cache 114 is to temporarily store the next one or more to-be-outputted transport packets pending output from the adaptor 110 or the last one or more transport packets recently received at the adaptor 110. The use of the cache 114 enables transport packets to be received and stored or to be retrieved and outputted with minimal latency (most notably without incurring transfer latency across the bus 130). The cache 114 also stores descriptor data for each transport packet" [0074], "The descriptors are maintained in order of receipt in the receipt queue. In the case of outputting transport packets from an output port, the data link control circuit sequentially retrieves each descriptor from the transmit queue, and the transport packet to which each retrieved descriptor points" [0037], "When an adaptor is used to input transport streams, the data link control circuit allocates to each received transport packet to be retained, an unused descriptor in one of a sequence of descriptor storage locations, of a queue allocated to the input port" [0035]);

a multiplexer for reading the instruction set and for multiplexing the plurality of video elementary data streams and the plurality of audio elementary data streams to generate a multiplexed stream (figs. 1-2), the multiplexer comprising:

a direct memory access (DMA) circuit (DMA control circuit 116, fig. 2) for connection to the bus (bus 130, fig. 2) for directly accessing the plurality of video and audio elementary data streams stored in the data memory and accessing the instruction set stored in the instruction memory, wherein each video and audio elementary data streams is stored as data units in the data memory, and the instruction set state the storage location of data units in the data memory and an order of multiplexing the corresponding data units ("The DMA control circuit 116 is for transferring transport packet data and descriptor data between the host memory 120 and the cache 114. The DMA control circuit 116 can maintain a sufficient number of transport packets (and descriptors therefor) in the cache 114 to enable the data link control circuit 112 to output transport packets in the output TS continuously (i.e., in successive time slots). The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots" [0076], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The

descriptors are retrieved from the beginning of the sequence in order" [0036]) and for storing the instruction set into the memory ("The cache 114 also stores descriptor data for each transport packet" [0074], "The memory can store one or more queues of descriptor storage locations, such as a queue assigned to an input port and a queue assigned to an output port" [0034]);

wherein the multiplexer is operable to generate a multiplexed stream by reading instruction set with said DMA circuit from the instruction memory, by reading data units from storage locations in a predetermined order stated in the read instruction set, and by outputting the read data units as said multiplexed stream ("When transmitting packets, the data link control circuit 112 retrieves descriptors for outgoing transport packets from the cache 114 and transmits the corresponding transport packets in time slots of the outgoing TS that occur when the time of the reference clock generator 113 approximately equals the dispatch times indicated in the respective descriptors" [0078] and "combine program information of two TSs, namely, TS1 and TS2, into a third TS, namely, TS3" [0093]; "TS1 illustratively is received at a first adaptor 110, TS2 illustratively is received at a second adaptor 110 and TS3 illustratively is transmitted from a third adaptor 110 of the same remultiplexer node 100" [0093]; "the processor 160 examines descriptors from the receipt queues (and/or possibly other queues containing descriptors of transport packets not yet scheduled for output) and identifies up to $j \geq 1$ descriptors pointing to transport packets to be outputted from the interrupting adaptor 110"; and further "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the

transport packet in the transport packet storage location to which the descriptor points" [0144]);

wherein the direct memory access (DMA) circuit is connected to said bus for directly accessing the plurality of video and audio elementary data streams stored in the data memory and directly accessing the multiplexing instruction data stored in the instruction memory (figs. 1-2).

Regarding claims **22, 32, 42**, Robinett discloses everything claimed as applied above (see claims 21, 31, 41). In addition, Robinett discloses the CPU generates the instruction set and command instruction data having stated therein an instruction for data processing to be executed in an arbitrary position in the instruction set, and stores the generated instruction set and command instruction data into the instruction memory ("The cache 114 also stores descriptor data for each transport packet" [0074]; and "The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076]; and "each receipt descriptor has a field 129-9 in which information pertinent to scrambling or descrambling can be stored, such as the control word to be used in scrambling the transport packet or a pointer to the appropriate control word table containing control words for use in scrambling or descrambling the transport packet" [0165]);

the multiplexer generates a multiplexed stream including the elementary data streams and command data by reading the instruction set and command instruction data from the instruction memory, reading the data units from the storage locations in the predetermined order stated in the read instruction set and outputting the read data units, when having read the instruction set ("combine program information of two TSs, namely, TS1 and TS2, into a third TS, namely, TS3" [0093]; "TS1 illustratively is received at a first adaptor 110, TS2 illustratively is received at a second adaptor 110 and TS3 illustratively is transmitted from a third adaptor 110 of the same remultiplexer node 100" [0093]; "the processor 160 examines descriptors from the receipt queues (and/or possibly other queues containing descriptors of transport packets not yet scheduled for output) and identifies up to j≥1 descriptors pointing to transport packets to be outputted from the interrupting adaptor 110"; and further "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the transport packet in the transport packet storage location to which the descriptor points" [0144]), or outputting command data having stated therein the execution instruction stated in the command instruction data, when having read the command instruction data ("In processing descriptors and transport packets, the descrambler 115 uses the PID of the transport packet, to which the currently examined descriptor points, to index a descrambling map located in the cache 114" [0172] and "The indexed entry of the descrambling map indicates whether or not the transport packet is scrambled and, if scrambled, one or more control words that can be used to descramble the transport packet. The indexed entry of the

descrambling map can contain the control words corresponding to the PID of the transport packet or a pointer to a memory location in which the respective control word is stored" [0172]); and

the apparatus further including a command executing means which is supplied with the multiplexed stream and performs processing corresponding to an instruction content stated in the command data when the data row in the input multiplexed stream is command data ("If the indexed entry of the descrambling map indicates that the transport packet is to be descrambled, the descrambler 115 obtains the control word corresponding to the PID of the transport packet and descrambles the transport packet data using the control word" [0172]), or outputs the input multiplexed stream as it is when the data row in the input multiplexed stream is elementary data stream ("If the indexed entry of the descrambling map indicates that the transport packet to which the accessed descriptor points is not to be descrambled, the descrambler 115 simply sets the status bit(s) 129-7 of the descriptor to indicate that the next processing step, according to the order of the defined sequence of processing steps, may be performed on the descriptor and transport packet to which it points" [0172]).

Regarding claims **3, 13, 23, 33, 43**, Robinett et al. discloses everything claimed as applied above (see claims 2, 12, 22, 32, 42 respectively). In addition, Robinett et al. discloses the multiplexed stream generating means outputs, synchronously with the multiplexed stream, an ID flag for identifying which data row in the multiplexed stream is command data or elementary data stream ("each receipt descriptor has a field 129-9 in which information pertinent to scrambling or descrambling can be stored, such as the

control word to be used in scrambling the transport packet or a pointer to the appropriate control word table containing control words for use in scrambling or descrambling the transport packet" [0165]);

the command executing means judges based on the ID flag whether the data row in the multiplexed stream is command data or elementary data stream ("In processing descriptors and transport packets, the descrambler 115 uses the PID of the transport packet, to which the currently examined descriptor points, to index a descrambling map located in the cache 114" [0172] and "If the indexed entry of the descrambling map indicates that the transport packet to which the accessed descriptor points is not to be descrambled, the descrambler 115 simply sets the status bit(s) 129-7 of the descriptor to indicate that the next processing step, according to the order of the defined sequence of processing steps, may be performed on the descriptor and transport packet to which it points. If the indexed entry of the descrambling map indicates that the transport packet is to be descrambled, the descrambler 115 obtains the control word corresponding to the PID of the transport packet and descrambles the transport packet data using the control word" [0172] and [0173]).

Regarding claims **7, 17, 27, 37, 47**, Robinett et al. discloses everything claimed as applied above (see claims 2, 12, 22, 32, 42 respectively). In addition, Robinett et al. discloses the instruction generating means generates, when sending a timing acknowledgment in an arbitrary timing in an output multiplexed stream, command instruction data having stated therein an instruction for sending a timing acknowledgment ("The field 129-5 is for storing the receipt time for an incoming

received transport packet or for storing the dispatch time of an outgoing to-be-transmitted transport packet" [0086]);

the multiplexed stream generating means outputs, when the command instruction data has stated therein an instruction for sending the timing acknowledgment, the command data having stated therein the content stated in the command instruction data ("The actual dispatch time is then stored in field 129-5 of the transmit descriptor. As described below, the actual dispatch time is really an approximate time at which the data link control circuit 112 of the third adaptor 110 (which outputs the remultiplexed TS TS3) submits the corresponding transport packet for output" [0138]); and

the command executing means sends, when the command data has stated therein an instruction for sending the timing acknowledgment, the timing acknowledgment in a position of the command data in the multiplexed stream ("When the time of the reference clock generator 113 of the third adaptor 110 equals the time indicated in the dispatch time field 129-5 of the retrieved descriptor, the data link control circuit 112 transmits the transport packet, to which the descriptor (in the storage location pointed to by the head pointer 124-3) points, in TS3" [0144]).

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 4-6, 14-16, 24-26, 34-36, 44-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinett et al. further in view of Kelly et al. (PG Pub US 2001/0036355 A1).

Regarding claims **4, 14, 24, 34, 44**, Robinett et al. discloses everything claimed as applied above (see claims 2, 12, 22, 32, 42 respectively). However, although Robinett et al. discloses "remultiplexing involves the selective modification of the content of a TS, such as adding transport packets to a TS" ([0025]), Robinett et al. fails to specifically disclose the instruction generating means generates, when inserting stuffing data into an output multiplexed stream, command instruction data having stated therein an instruction for inserting the stuffing data and an amount of the stuffing data; the multiplexed stream generating means outputs, when the command instruction data has stated therein an instruction for inserting the stuffing data, the command data having stated therein the content stated in the command instruction data; and the command executing means inserts, when the command data has stated therein an instruction for inserting the stuffing data, stuffing data whose amount is stated in the command data to a position of the command data in the multiplexed stream.

Nevertheless, Kelly et al. teaches "depending on the contents of field AFC, there may be present an adaptation field AF, occupying some of the space otherwise allocated to payload data. The adaptation field AF may for example contain a discontinuity indicator flag as defined in ISO/IEC 13818 for MPEG2. When set to `1`, this flag indicates that the discontinuity state is true for the current Transport Stream packet. The discontinuity indicator is used to indicate two types of discontinuities,

system time-base discontinuities and continuity counter discontinuities. In addition to optional data fields of pre-defined meaning, the adaptation field can be padded with stuffing bytes, so as to match the PES packet ends to TS packet boundaries" (Kelly et al. [0061]) and "stuffing can be done by either adding a PES stuffing packet or by adding an adaptation field. The adaptation field allows any desired number of data bytes to be added to the PES packet, as described in the MPEG specification. The data can be meaningless for stuffing purposes" (Kelly et al. [0122]).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to generate in the instruction generating means, when inserting stuffing data into an output multiplexed stream, command instruction data having stated therein an instruction for inserting the stuffing data and an amount of the stuffing data; output in the multiplexed stream generating means, when the command instruction data has stated therein an instruction for inserting the stuffing data, the command data having stated therein the content stated in the command instruction data; and insert in the command executing means, when the command data has stated therein an instruction for inserting the stuffing data, stuffing data whose amount is stated in the command data to a position of the command data in the multiplexed stream because "audio frames are not aligned with Transport packets, it may be necessary to stuff part of the last audio packet to remove the start of the next audio frame" (Kelly et al. [0122]).

Regarding claims **5, 15, 25, 35, 45**, Robinett et al. discloses everything claimed as applied above (see claims 2, 12, 22, 32, 42 respectively). However, although

Robinett et al. discloses “remultiplexing involves the selective modification of the content of a TS, such as deleting transport packets from a TS” ([0025]), Robinett et al. fails to specifically disclose the instruction generating means generates, when deleting data from an output multiplexed stream, command instruction data having stated therein a data delete instruction and data amount to be deleted; the multiplexed stream generating means outputs, when the command instruction data has stated therein an instruction for deletion of data, the command data having stated therein the content stated in the command instruction data; and the command executing means deletes, when the command data has stated therein an instruction for deletion of the data, an amount of data stated in the command data from a multiplexed stream next to the command data.

Nevertheless, Kelly et al. teaches “To avoid problems with the audio buffer model, it may be necessary to delete some audio packets. After the end of the last video frame in the first stream SEQ1, once the start of a new audio frame is found, that packet and all subsequent audio packets should be deleted before sending over the digital interface. Conventional null TS packets can be inserted in their place, to preserve the TS format. Leading audio (i.e. audio that precedes the first video packet) is deleted similarly to avoid problems with audio buffer overflow” (Kelly et al. [0157]).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to generate in the instruction generating means, when deleting data from an output multiplexed stream, command instruction data having stated therein a data delete instruction and data amount to be deleted; output in

the multiplexed stream generating means, when the command instruction data has stated therein an instruction for deletion of data, the command data having stated therein the content stated in the command instruction data; and delete in the command executing means, when the command data has stated therein an instruction for deletion of the data, an amount of data stated in the command data from a multiplexed stream next to the command data because “the null packets may be deleted by re-multiplexing processes and, therefore, the delivery of the payload of null packets to the decoder cannot be assumed” (Kelly et al. [0138]).

Regarding claims **6, 16, 26, 36, 46**, Robinett et al. discloses everything claimed as applied above (see claims 2, 12, 22, 32, 42 respectively). However, although Robinett et al. discloses “remultiplexing involves the selective modification of the content of a TS, such as adding transport packets to a TS” ([0025]), Robinett et al. fails to specifically disclose the instruction generating means generates, when inserting arbitrary data into an output multiplexed stream, command instruction data having stated therein an instruction for insertion of the arbitrary data; the multiplexed stream generating means outputs, when the command instruction data has stated therein an instruction for insertion of the arbitrary data, the command data having stated therein the content stated in the command instruction data; and the command executing means inserts, when the command data has stated therein an instruction for insertion of the arbitrary data, the arbitrary data stated in the command data to a position of the command data in the multiplexed stream.

Nevertheless, Kelly et al. teaches “depending on the contents of field AFC, there may be present an adaptation field AF, occupying some of the space otherwise allocated to payload data. The adaptation field AF may for example contain a discontinuity indicator flag as defined in ISO/IEC 13818 for MPEG2. When set to `1`, this flag indicates that the discontinuity state is true for the current Transport Stream packet. The discontinuity indicator is used to indicate two types of discontinuities, system time-base discontinuities and continuity counter discontinuities. In addition to optional data fields of pre-defined meaning, the adaptation field can be padded with stuffing bytes, so as to match the PES packet ends to TS packet boundaries” (Kelly et al. [0061]) and “stuffing can be done by either adding a PES stuffing packet or by adding an adaptation field. The adaptation field allows any desired number of data bytes to be added to the PES packet, as described in the MPEG specification. The data can be meaningless for stuffing purposes” (Kelly et al. [0122]).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to generate in the instruction generating means, when inserting arbitrary data into an output multiplexed stream, command instruction data having stated therein an instruction for insertion of the arbitrary data; output in the multiplexed stream generating means, when the command instruction data has stated therein an instruction for insertion of the arbitrary data, the command data having stated therein the content stated in the command instruction data; and insert in the command executing means, when the command data has stated therein an instruction for insertion of the arbitrary data, the arbitrary data stated in the command data to a

position of the command data in the multiplexed stream because “audio frames are not aligned with Transport packets, it may be necessary to stuff part of the last audio packet to remove the start of the next audio frame” (Kelly et al. [0122]).

5. Claims 8-9, 18-19, 28-29, 38-39, 48-49 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinett et al. further in view of Dobson et al. (US Patent No. 6,188,703 B1).

Regarding claims **8, 18**, Robinett et al. discloses a multiplexing apparatus which multiplexes a plurality of video elementary data streams and a plurality of audio elementary data streams to generate one multiplexed stream (fig. 1), the multiplexing apparatus comprising:

encoders receiving video data from a video source, receiving audio data from an audio source, encoding the video data into the video elementary data streams, encoding the audio data into the audio elementary data streams, dividing the video elementary data streams into a plurality of video data units and dividing the audio elementary streams into a plurality of audio data units (“TSs are bit streams that contain the data of one or more compressed/encoded audiovideo programs. Each TS is formed as a sequence of fixed length transport packets. Each compressed program includes data for one or more compressed elementary streams (ESs), such as a digital video signal and/or a digital audio signal. The transport packets also carry program clock references (PCRs) for each program, which are time stamps of an encoder system time clock to which the decoding and presentation of the respective program is synchronized” [0033], “one or more data injection sources 50 and one or more data

extraction destinations 60. These sources 50 and destinations 60 may themselves be implemented as PC compatible computers. However, the sources 50 may also be devices such as cameras, video tape players, communication demodulators/receivers and the destinations may be display monitors, video tape recorders, communications modulators/transmitters, etc. The data injection sources 50 supply TS, ES or other data to the remultiplexer 30, e.g., for remultiplexing into the outputted TSs TS4 and/or TSS" [0066]);

a memory which stores the plurality of video and audio data units that are composed of an arbitrary amount of said video and audio elementary data streams ("The use of the cache 114 enables transport packets to be received and stored or to be retrieved and outputted" [0074] and further "TSs are bit streams that contain the data of one or more compressed/encoded audio-video programs. Each TS is formed as a sequence of fixed length transport packets" [0033]);

a counting means for indicating a count which indicates a data occupancy of the memory ("A queue is implemented in each ring 124 by designating a pointer 124-3 to a head of the queue or first used/allocated descriptor storage location 129 in the queue and a pointer 124-4 to a tail of the queue or last used/allocated descriptor storage location 129 in the queue" [0081]);

an instruction generating means for calculating an order of multiplexing the video data units and audio data units based on storage locations supplied by the encoders ("The descriptors are maintained in order of receipt in the receipt queue. In the case of outputting transport packets from an output port, the data link control circuit sequentially

retrieves each descriptor from the transmit queue, and the transport packet to which each retrieved descriptor points" [0037], "When an adaptor is used to input transport streams, the data link control circuit allocates to each received transport packet to be retained, an unused descriptor in one of a sequence of descriptor storage locations, of a queue allocated to the input port" [0035]), generating an instruction set of a plurality of multiplexing instruction data which describe the storage location and order of multiplexing of each data unit and storing the instruction set into the memory ("The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the sequence in order" [0036]); and

a multiplexed stream generating means for reading the instruction set ("When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the sequence in order" [0036]), for generating one multiplexed stream by reading the data units from the memory in a predetermined order based upon the instruction set ("The

data link control circuit also retrieves from the cache the transport packers stored in transport packet storage locations to which the retrieved descriptors point. The data link control circuit outputs each retrieved transport packet in a unique time slot (i.e., one transport packet per time slot) of a transport stream outputted from the output port. The direct memory access circuit obtains from the memory for storage in the cache, descriptors of the queue assigned to the output port in storage locations following the descriptor storage locations in which a last cached descriptor of the sequence is stored. The direct memory access circuit also obtains each transport packet stored in a transport packet location to which the obtained descriptors point" [0036]) and outputting the data units corresponding to the instruction set ("combine program information of two TSs, namely, TS1 and TS2, into a third TS, namely, TS3" [0093]; "TS1 illustratively is received at a first adaptor 110, TS2 illustratively is received at a second adaptor 110 and TS3 illustratively is transmitted from a third adaptor 110 of the same remultiplexer node 100" [0093]; "the processor 160 examines descriptors from the receipt queues (and/or possibly other queues containing descriptors of transport packets not yet scheduled for output) and identifies up to j≥1 descriptors pointing to transport packets to be outputted from the interrupting adaptor 110"; and further "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the transport packet in the transport packet storage location to which the descriptor points" [0144]);

the instruction generating means adding a data amount of a data unit corresponding to the multiplexing instruction data to the count ("The field 129-8 contains

a transfer count indicating the number of bytes in a received incoming transport packet" [0089] and further "the field 129-3 is for storing the number of bytes of a to-be-outputted, outgoing transport packet" [0084]).

However, Robinett et al. fails to specifically disclose that the instruction generating means adds a data amount of a data unit corresponding to the multiplexing instruction data to the count and the counting means subtracts the data amount of output data unit from the count.

Nevertheless, Dobson et al. teaches "a video FIFO fullness counter 40 (see FIG. 4) then keeps track of the number of bytes of video data in the FIFO 32 at any time" (Dobson et al. column 4 lines 3-6) and "when a start code is detected, the value in the FIFO-fullness-counter 40 is latched into another counter called the start-code position counter 48. The start-code position counter 48 only counts down on compressed data FIFO reads by the mux 30" (Dobson et al. column 4 lines 31-35).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to add a data amount of a data unit corresponding to the multiplexing instruction data to the count and subtract the data amount of output data unit from the count because "a count of the start-of-frame can be used to determine when to insert the PTS" (Dobson et al. column 5 lines 27-29).

Regarding claims **28, 38, 48**, Robinett discloses everything claimed as applied above (see claims 21, 31, 41). In addition, Robinett disclose a counting means in the multiplexer for indicating a count which indicates a data occupancy of the memory ("A queue is implemented in each ring 124 by designating a pointer 124-3 to a head of the

queue or first used/allocated descriptor storage location 129 in the queue and a pointer 124-4 to a tail of the queue or last used/allocated descriptor storage location 129 in the queue" [0081]);

wherein the CPU adds the data amount of a data unit corresponding to the generated instruction set to the count ("The field 129-8 contains a transfer count indicating the number of bytes in a received incoming transport packet" [0089] and further "the field 129-3 is for storing the number of bytes of a to-be-outputted, outgoing transport packet" [0084]).

However, Robinett fails to specifically disclose the CPU adds the data amount of a data unit corresponding to the generated multiplexing instruction data to the count and the counting means subtracts the data amount of output data unit from the count.

Nevertheless, Dobson et al. teaches "a video FIFO fullness counter 40 (see FIG. 4) then keeps track of the number of bytes of video data in the FIFO 32 at any time" (Dobson et al. column 4 lines 3-6) and "when a start code is detected, the value in the FIFO-fullness-counter 40 is latched into another counter called the start-code position counter 48. The start-code position counter 48 only counts down on compressed data FIFO reads by the mux 30" (Dobson et al. column 4 lines 31-35).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have the CPU add the data amount of a data unit corresponding to the generated multiplexing instruction data to the count and the counting means subtract the data amount of output data unit from the count because "a

count of the start-of-frame can be used to determine when to insert the PTS" (Dobson et al. column 5 lines 27-29).

Regarding claims **9, 19, 29, 39, 49**, Robinett et al. and Dobson et al. discloses everything claimed as applied above (see claims 8, 18, 28, 38, 48 respectively). However, Robinett et al. fails to disclose that the memory is divided into a plurality of storage areas correspondingly to the types of the elementary data streams and the elementary data streams is stored into corresponding storage areas; the counting means holds a plurality of counts corresponding to the storage areas in the memory; the instruction generating means adds the data amount of a data unit corresponding to the generated instruction set to a count corresponding to a storage area in which the data unit is stored; and the counting means subtracts the data amount of data unit output from the memory from a count corresponding to the storage area in which the data unit is stored.

Nevertheless, Dobson et al. teaches "a video FIFO fullness counter 40 (see FIG. 4) then keeps track of the number of bytes of video data in the FIFO 32 at any time" (Dobson et al. column 4 lines 3-6) and "when a start code is detected, the value in the FIFO-fullness-counter 40 is latched into another counter called the start-code position counter 48. The start-code position counter 48 only counts down on compressed data FIFO reads by the mux 30" (Dobson et al. column 4 lines 31-35) and "the audio mux logic circuit 18 performs similar functions to the video mux logic circuit 16. It buffers the audio elementary stream data, signals the mux processor 22 when there is sufficient audio data in the FIFO to form the payload of a MPEG-2 transport packet and signals

the mux processor 22 when the data in the current payload contains an audio start-of-frame. Audio logic is implemented in exactly the same way as the video logic and is implemented in the same logic block" (Dobson et al. column 4 lines 48-56).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to divide the memory into a plurality of storage areas correspondingly to the types of the elementary data streams and to store the elementary data streams into corresponding storage areas; the counting means to hold a plurality of counts corresponding to the storage areas in the memory; the instruction generating means to add the data amount of a data unit corresponding to the generated instruction set to a count corresponding to a storage area in which the data unit is stored; and the counting means to subtract the data amount of data unit output from the memory from a count corresponding to the storage area in which the data unit is stored because "a count of the start-of-frame can be used to determine when to insert the PTS" (Dobson et al. column 5 lines 27-29).

6. Claims 10, 20, 30, 40, 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Robinett et al. further in view of Zaun et al. (PG Pub US 2001/0024456 A1).

Regarding claims **10 and 20**, Robinett et al. discloses a multiplexing apparatus which multiplexes a plurality of video elementary data streams and a plurality of audio elementary data streams to generate a plurality of multiplexed streams (fig. 1), the multiplexing apparatus comprising:

encoders receiving video data from a video source, receiving audio data from an audio source, encoding the video data into the video elementary data streams, encoding the audio data into the audio elementary data streams, dividing the video elementary data streams into a plurality of video data units and dividing the audio elementary streams into a plurality of audio data units ("TSs are bit streams that contain the data of one or more compressed/encoded audiovideo programs. Each TS is formed as a sequence of fixed length transport packets. Each compressed program includes data for one or more compressed elementary streams (ESs), such as a digital video signal and/or a digital audio signal. The transport packets also carry program clock references (PCRs) for each program, which are time stamps of an encoder system time clock to which the decoding and presentation of the respective program is synchronized" [0033], "one or more data injection sources 50 and one or more data extraction destinations 60. These sources 50 and destinations 60 may themselves be implemented as PC compatible computers. However, the sources 50 may also be devices such as cameras, video tape players, communication demodulators/receivers and the destinations may be display monitors, video tape recorders, communications modulators/transmitters, etc. The data injection sources 50 supply TS, ES or other data to the remultiplexer 30, e.g., for remultiplexing into the outputted TSs TS4 and/or TSS" [0066]);

a memory which stores the plurality of video and audio data units that are composed of arbitrary amounts of said video and audio elementary data streams ("The use of the cache 114 enables transport packets to be received and stored or to be

retrieved and outputted" [0074] and further "TSs are bit streams that contain the data of one or more compressed/encoded audio-video programs. Each TS is formed as a sequence of fixed length transport packets" [0033]);

an instruction generating means for calculating an order of multiplexing the video data units and audio data units based on storage locations supplied by the encoders ("The descriptors are maintained in order of receipt in the receipt queue. In the case of outputting transport packets from an output port, the data link control circuit sequentially retrieves each descriptor from the transmit queue, and the transport packet to which each retrieved descriptor points" [0037], "When an adaptor is used to input transport streams, the data link control circuit allocates to each received transport packet to be retained, an unused descriptor in one of a sequence of descriptor storage locations, of a queue allocated to the input port" [0035]), generating an instruction set of a plurality of multiplexing instruction data which describe the storage location and order of multiplexing of each data unit ("The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the

sequence in order" [0036]) and storing the instruction set into the memory ("The cache 114 also stores descriptor data for each transport packet" [0074] and "The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076]); and

a multiplexed stream generating means for generating a plurality of multiplexed streams by reading the data units in a predetermined order based upon the instruction set from the memory ("The data link control circuit also retrieves from the cache the transport packers stored in transport packet storage locations to which the retrieved descriptors point. The data link control circuit outputs each retrieved transport packet in a unique time slot (i.e., one transport packet per time slot) of a transport stream outputted from the output port. The direct memory access circuit obtains from the memory for storage in the cache, descriptors of the queue assigned to the output port in storage locations following the descriptor storage locations in which a last cached descriptor of the sequence is stored. The direct memory access circuit also obtains each transport packet stored in a transport packet location to which the obtained descriptors point" [0036]), and outputting the data units corresponding to the instruction set ("one or more to-be-remultiplexed TSs, namely, TS1, TS2 and TS3, are received at the remultiplexer 30. As a result of the remultiplexing operation of the remultiplexer 30,

one or more TSs, namely, TS4 and TS5, are outputted from the remultiplexer 30" [0065]; "the processor 160 examines descriptors from the receipt queues (and/or possibly other queues containing descriptors of transport packets not yet scheduled for output) and identifies up to j≥1 descriptors pointing to transport packets to be outputted from the interrupting adaptor 110"; and further "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the transport packet in the transport packet storage location to which the descriptor points" [0144])

the instruction generating means stating, in the multiplexing instruction data, the type of a multiplexed stream resulted from multiplexing data units corresponding to the generated multiplexing instruction data ("Each TS is provided with a four byte header that includes a packet identifier or "PID." The PID is analogous to a tag which uniquely indicates the contents of the transport packet. Thus, one PID is assigned to a video ES of a particular program, a second, different PID is assigned to the audio ES of a particular program, etc" [0018]).

However, Robinett et al. fails to specifically disclose the multiplexed stream generating means generating the plurality of multiplexed streams by switching the outputting of the data unit read correspondingly to the multiplexed stream type stated in the multiplexing instruction data.

Nevertheless, Zaun et al. teaches "the output processor 124 is a FPGA conducts the required hardware tasks to generate two or more output streams from the data stored in the packet buffers 104. More particularly, the output processor 124 reads the

selected packet data from the input packet buffers and/or the insert packet buffer 112" (Zaun et al. [0035]) and "the output processing section then generates two or more independent high-speed transport multiplex (HSTM) output streams incorporating the selected packet data" (Zaun et al. [0035]) and "the bus control logic 400 responds to packet buffer interrupts, reads new packet information from the interrupting packet buffers, and generates addresses and chip selects to access data in the packet buffer as instructed by the output stream data registers. The chip selects in particular are used by the output processor 124 to read selected individual "chips" in the packet buffers 104" (Zaun et al. [0036]).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to generate the plurality of multiplexed streams by switching the outputting of the data unit read correspondingly to the multiplexed stream type stated in the multiplexing instruction data because "there are some applications where it is desirable to generate two or more output streams from the input streams" (Zaun et al. [0003]).

Regarding claims **30, 40**, **50**, Robinett discloses everything claimed as applied above (see claims 21, 31, 41). In addition, Robinett discloses the multiplexer generates a plurality of multiplexed streams by reading the generated instruction set from the instruction memory, reading the data units from the storage locations stated in the read instruction set and by outputting the read data units ("The cache 114 also stores descriptor data for each transport packet" [0074] and "The DMA control circuit 116 is for transferring transport packet data and descriptor data between the hose memory 120

and the cache 114. the DMA control circuit 116 can maintain a sufficient number of transport packets (and descriptors therefor) in the cache 114 to enable the data link control circuit 112 to output transport packets in the output TS continuously (i.3., in successive time slots). The DMA control circuit 116 can also obtain control of a sufficient number of descriptor storage locations, and the packet storage locations to which they point, in the cache 114. The DMA control circuit 116 obtains control of such descriptor and transport packet storage locations for the cache 114. This enables continuous allocation of descriptors and transport packet storage locations to incoming transport packets as they are received (i.e., from successive time slots)" [0076]);

the CPU states, in the instruction set, the type of a multiplexed stream resulted from multiplexing data units corresponding to the generated instruction set ("Each TS is provided with a four byte header that includes a packet identifier or "PID." The PID is analogous to a tag which uniquely indicates the contents of the transport packet. Thus, one PID is assigned to a video ES of a particular program, a second, different PID is assigned to the audio ES of a particular program, etc" [0018]); and

However, Robinett fails to specifically disclose the multiplexer generates the plurality of multiplexed streams by switching the outputting of the read data unit according to the multiplexed stream type stated in the read instruction set.

Nevertheless, Zaun et al. teaches "the output processor 124 is a FPGA conducts the required hardware tasks to generate two or more output streams from the data stored in the packet buffers 104. More particularly, the output processor 124 reads the selected packet data from the input packet buffers and/or the insert packet buffer 112"

(Zaun et al. [0035]) and “the output processing section then generates two or more independent high-speed transport multiplex (HSTM) output streams incorporating the selected packet data” (Zaun et al. [0035]) and “the bus control logic 400 responds to packet buffer interrupts, reads new packet information from the interrupting packet buffers, and generates addresses and chip selects to access data in the packet buffer as instructed by the output stream data registers. The chip selects in particular are used by the output processor 124 to read selected individual "chips" in the packet buffers 104” (Zaun et al. [0036]).

Therefore, it would have been obvious to a person having ordinary skill in the art at the time the invention was made to have the multiplexer generates the plurality of multiplexed streams by switching the outputting of the read data unit according to the multiplexed stream type stated in the read instruction set because “there are some applications where it is desirable to generate two or more output streams from the input streams” (Zaun et al. [0003]).

Response to Arguments

Previous minor informality objection to claims 1, 2, 8, 10, 21, 31 are withdrawn in view of Applicant’s amendment.

Applicant's arguments have been considered but are moot in view of the new ground(s) of rejection.

Applicant's arguments have been fully considered but they are not persuasive.

Applicants have argued regarding claims 1, 2, 8, 10, 11, 12, 18, 20, 21, 31, 41 that "nothing in Robinett et al. shows, teaches or suggests calculating an order of multiplexing based on storage locations (supplied by encoders)" (pages 34, 37, 38).

In response to Applicants' argument, the examiner respectfully disagrees. Robinett discloses "The direct memory access circuit obtains control of one or more unused descriptor storage locations of the queue in the memory following a last descriptor storage location of which the cache has already obtained control. The direct memory access circuit also obtains control of transport packet locations in the memory to which such descriptors in the one or more descriptor storage locations point" [0035], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of. descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the sequence in order. The data link control circuit also retrieves from the cache the transport packers stored in transport packet storage locations to which the retrieved descriptors point" [0036], "The descriptors are maintained in order of receipt in the receipt queue. In the case of outputting transport packets from an output port, the data link control circuit sequentially retrieves each descriptor from the transmit queue, and the transport packet to which each retrieved descriptor points. At a time corresponding to a dispatch time recorded in each retrieved descriptor, the data link control circuit transmits the retrieved transport packet to which each retrieved descriptor points in a time slot of the outputted transport stream corresponding to the dispatch time recorded in the retrieved descriptor" [0037]), "A queue is implemented in each ring 124 by

designating a pointer 124-3 to a head of the queue or first used/allocated descriptor storage location 129 in the queue and a pointer 124-4 to a tail of the queue or last used/allocated descriptor storage location 129 in the queue. Descriptor storage locations 129 are allocated for incoming transport packets starting with the unused/non-allocated descriptor storage location 129 immediately following the tail 124-4. Descriptor storage locations 129 for outgoing transport packets are retrieved from the queue starting from the descriptor storage location 129 pointed to by the head 124-3 and proceeding in sequence to the tail 124-4" [0081]). This shows that as the node receives transport packets, an order is sequentially maintained in the queue of each ring. Therefore, Robinett discloses calculating an order of multiplexing based on storage locations.

Applicants have argued regarding claims 1, 2, 8, 10, 11, 12, 18, 20, 21, 31, 41 that "nothing in Robinett et al. shows, teaches or suggests generating an instruction set which describes the storage location and (calculated) order of multiplexing" (pages 34, 37, 38).

In response to Applicants' argument, the examiner respectfully disagrees. Robinett discloses "Each ring 124 is a sequence of descriptor storage locations 129 from a starting memory address or top of ring 124-1 to an ending memory address or bottom of ring 124-2. One ring 124 is provided for each outgoing TS transmitted from the remultiplexer node 100 and one ring 124 is provided for each incoming TS received at the remultiplexer node 100. Other rings 124 may be provided as described in greater detail below" [0080], "A queue is implemented in each ring 124 by designating a pointer

124-3 to a head of the queue or first used/allocated descriptor storage location 129 in the queue and a pointer 124-4 to a tail of the queue or last used/allocated descriptor storage location 129 in the queue. Descriptor storage locations 129 are allocated for incoming transport packets starting with the unused/non-allocated descriptor storage location 129 immediately following the tail 124-4. Descriptor storage locations 129 for outgoing transport packets are retrieved from the queue starting from the descriptor storage location 129 pointed to by the head 124-3 and proceeding in sequence to the tail 124-4" [0081], "The field 129-4 is for storing a pointer to the transport packet storage location to which the descriptor corresponds. This is illustrated in FIG. 2 by use of arrows from the descriptors in descriptor storage locations 129 in the ring 124 to specific storage locations of the transport packet pool 122" [0085]). This shows that the rings maintain the calculated order in the queues as well as the storage location in the descriptor storage location. Therefore, Robinett discloses generating an instruction set which describes the storage location and (calculated) order of multiplexing.

Applicants have argued regarding claims 1, 2, 8, 10, 11, 12, 18, 20, 21, 31, 41 that "nothing Robinett et al. shows, teaches or suggests reading the instruction set from a memory and generating a multiplexed stream by reading the data units from the memory in a predetermined order based on the instruction set" (pages 35, 37, 38).

In response to Applicants' argument, the examiner respectfully disagrees. Robinett discloses "Descriptor storage locations 129 for outgoing transport packets are retrieved from the queue starting from the descriptor storage location 129 pointed to by

the head 124-3 and proceeding in sequence to the tail 124-4. Whenever the descriptor of the descriptor storage location 129 at the end of the ring 124-2 is reached, allocation or retrieval of descriptors from descriptor storage locations 129 continues with the descriptor of the descriptor storage location 129 at the top of the ring 124-1" [0081], "The data link control circuit 112 sequentially retrieves from the cache 114 each descriptor in the transmit queue, in order from the head pointer 124-3, and the transport packet in the transport packet storage location to which the descriptor points. When the time of the reference clock generator 113 of the third adaptor 110 equals the time indicated in the dispatch time field 129-5 of the retrieved descriptor, the data link control circuit 112 transmits the transport packet, to which the descriptor (in the storage location pointed to by the head pointer 124-3) points, in TS3" [0144], "When an adaptor is used to output transport packets, the data link control circuit retrieves from the cache each descriptor of a sequence of. descriptor storage locations of a queue assigned to the output port. The descriptors are retrieved from the beginning of the sequence in order. The data link control circuit also retrieves from the cache the transport packers stored in transport packet storage locations to which the retrieved descriptors point. The data link control circuit outputs each retrieved transport packet in a unique time slot (i.e., one transport packet per time slot) of a transport stream outputted from the output port. The direct memory access circuit obtains from the memory for storage in the cache, descriptors of the queue assigned to the output port in storage locations following the descriptor storage locations in which a last cached descriptor of the sequence is stored. The direct memory access circuit also obtains each transport packet stored in a

transport packet location to which the obtained descriptors point" [0036]. This shows that the queue in the ring is read in order to generate a stream of the transport packets from which the descriptor storage location point to. Therefore, Robinett discloses reading the instruction set from a memory and generating a multiplexed stream by reading the data units from the memory in a predetermined order based on the instruction set.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTINE DUONG whose telephone number is (571)270-1664. The examiner can normally be reached on Monday - Friday: 830 AM-6 PM EST with first Friday off.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Seema Rao can be reached on (571) 272-3174. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Christine Duong/

Examiner, Art Unit 2462

04/14/2010

/Seema S. Rao/

Supervisory Patent Examiner, Art Unit 2462